Université Paris Diderot **Langages à objets avancés**
M1 **Année 2019-2020**

TP n° 10 : exceptions

Exercice 1 Matrices

1. Définir une classe `Matrice` qui représente une matrice d'entiers avec ses dimensions, nombre de lignes et de colonnes, et ses valeurs qui seront stockées dans un vecteur de vecteurs (`vector<vector<int>>`). Afin que l'écrasement de valeurs de la matrice soit contrôlé, on ajoute un attribut booléen `proteg` qui indique si les modifications sont possibles. Vous définirez également un constructeur prenant en paramètre les dimensions de la matrice.
2. Surcharger l'opérateur `<<` afin de pouvoir afficher une matrice.
3. Écrire une méthode `set(const int i, const int j, const int v)` qui attribue la valeur `v` à l'élément `i,j` de la matrice. Si la matrice est protégée en écriture la méthode devra propager une erreur de type `MatricePleine`. Il vous faudra donc créer la classe `MatricePleine` héritant de la classe `exception`. Cette classe aura trois attributs entiers et un constructeur à trois paramètres, la ligne et la colonne de l'élément à modifier, ainsi que la valeur qui aurait dû etre affectée. Il faudra également surchager l'opérateur `<<` afin de pouvoir afficher les renseignements sur l'erreur.
4. Les valeurs de la matrice doivent être compris entre 0 et 255. Modifier la méthode `set` afin qu'elle propage une erreur de type `DebordementVal` si la valeur à affecter n'est pas dans l'intervalle admissible. La classe `DebordementVal` aura comme pour `MatricePleine`, un constructeur à trois paramètres, la ligne et la colonne de l'élément à modifier, ainsi que la valeur qui aurait dû etre affectée. Instaurer alors une hiérarchie entre les classes d'exceptions (pensez à créer une nouvelle classe !). Il faudra de nouveau pouvoir afficher les renseignements sur les erreurs en utilisant l'opérateur `<<`.
5. Écrire une méthode `saisie` qui demande à l'utilisateur de remplir une matrice. Cette méthode devra invoquer la méthode `set`. Lorsque les valeurs entrées par l'utilisateur seront strictement inférieures à 0 (resp. supérieures à 255), la valeur sera mise à 0 (resp. 255). Si la matrice est protégée en écriture, il n'y aura pas de saisie et l'erreur `MatricePleine` sera repropagée.
6. Écrire une méthode `symetrise` qui copie dans la partie triangulaire inférieure de la matrice, la partie triangulaire supérieure. Si la matrice n'est pas carrée, la méthode propagera une erreur de type `ErreurDimMatrice`. La classe `ErreurDimMatrice` aura un attribut de type `string` et un constructeur à un paramètre représentant un message. Il faudra également surchager l'opérateur `<<` afin de pouvoir afficher le message de l'erreur.
7. Faire les tests nécessaires en faisant attention à la gestion des erreurs.